

## SAFE APPLICATION DISTRIBUTION AND EXECUTION IN A WIRELESS ENVIRONMENT

### Field of the Invention

[0001] The present invention relates to processing of applications for use in a wireless device, and more particularly, to increasing the security, safety and integrity of applications executed on a wireless device.

### Background

[0002] Wireless communication has experienced explosive growth in recent years. As consumers and businesses rely more on their wireless devices, such as mobile phones and personal digital assistants (PDAs), wireless service providers, i.e., carriers, strive to provide additional functionality on these wireless devices. This additional functionality would not only increase the demand for wireless devices but also increase the usage among current users. Increasing functionality, specifically by increasing the applications accessible by the wireless device, however, is costly and complicated thereby discouraging carriers from providing this functionality.

[0003] Furthermore, there is little to no assurance that an application, once placed on a wireless device, will execute properly. Currently, reliance on the application's ability to execute on a wireless device rest on the developer, the wireless device maker and/or the carrier. As more applications are developed and the number of applications on a wireless device increases, the wireless device environment becomes more dynamic. For example, a wireless device may choose to retrieve or execute a number of different applications from large pool of available applications at any given time. Thus, ensuring that any given application will be distributed to the wireless device and execute safely becomes much more difficult to control.

[0004] This is of particular concern because improper execution of an application may not only detrimentally affect the wireless device, but it may also be harmful to the carrier network and other network components, including other wireless devices. For example, one application, if not restricted, could take control of a wireless device's power control and cause interference among other wireless devices and decrease the overall capacity in the cell servicing the wireless device.

[0005] Currently, neither wireless device manufacturers nor carriers are equipped to support the testing and safe distribution of applications in a dynamic application distribution and execution environment. Thus, there is a concern that applications will be distributed and executed on wireless devices that may cause harm to the wireless device, carrier network, or other network components.

[0006] In addition, other safety issues arise as more applications are developed and the environment by which applications are transmitted to a wireless device becomes more dynamic. As the number of applications and the number of developers creating these applications increases, the desire to know the source of any given application, i.e., the developer, also increases. A carrier or a handset manufacturer will want to know, with some degree of reliability, that they can determine the source of an application should the application cause harm.

[0007] Consequently, what is needed in the art is a system and method for providing a more safe environment for the distribution and execution of applications on a wireless device.

### **Summary of the Invention**

[0008] Systems and methods consistent with the present invention overcome the shortcomings of existing systems by creating a safer environment for application distribution and execution that test applications with predetermined standards, provide traceability to the developer for nonrepudiation, check for unintended modifications to the application, allow the removal of the application from the wireless device, and/or use rules and permissions that define the environment on which an application may execute.

[0009] Certifying that an application meets predetermined standards provides the advantage of catching possible errors that could occur during execution ahead of time. This helps to prevent the detrimental effect of an application's execution.

[0010] Traceability provides the advantage of non-repudiation. If there is any problem with the application, it is beneficial to trace back to the source of the application, i.e., the developer, to correct the problem. In addition, having traceability discourages developers from creating applications that have harmful results, either whether intended or unintended.

[0011] Furthermore, the ability to determine if an application is modified prior to receiving it at the wireless device provides the advantage of increased safety by ensuring that the application received is the same one that was transmitted. As applications are distributed more freely in the wireless environment, the ability to determine if an application was modified increases the confidence that an application received by the wireless device was not modified, either accidentally or intentionally.

[0012] Providing a set of rules and permissions that define when applications may execute also increases the safety of an application distribution and execution system by preventing the unauthorized execution of an application on platforms, e.g., systems or environments in which it is not authorized.

[0013] The ability to remove applications from a wireless device also increases the safety of an application distribution system. If an application is installed on a handset, either by the manufacturer or through an application download, having a mechanism to remove the application because of unforeseen negative consequences increases the safety of an application distribution and execution system by removing harmful and undesirable code that could be detrimental.

[0014] Systems and methods consistent with the present invention may invoke one or more of the techniques disclosed herein. By invoking all of the techniques disclosed and referenced herein, however, systems and methods consistent with the present invention provide for high quality and safe distribution and execution of applications.

[0015] In one embodiment of the present invention, a method for distributing and processing an application, comprises the steps of receiving the application and identification information, certifying the application satisfies a predetermined criterion, assigning a permission to the application, transmitting the application, the permission and the identification information to the a device using a modification detection technique, determining whether the application was modified during transmission, storing a rule on the device, determining if the application may be process using the permission and the rule, and removing the application from the device.

[0016] In another embodiment of the present invention, a method for executing an application on a wireless device, comprises the steps of storing a rule to evaluate a permission, receiving information comprising the application, the permission and an

identification using a modification detection technique, receiving a request to execute the application on the wireless device, evaluating the received information to determine if the received information was modified, in the event the received information was not modified, evaluating the permission associated with the application, and in the event the permission is granted, executing the application.

[0017] In yet another embodiment of the present invention, a method for executing an application on a wireless device comprises the steps of storing a rule to evaluate a permission, receiving information comprising the application, the permission and an identification using a modification detection technique, receiving a request to execute the application on the wireless device, evaluating the received information to determine if the received information was modified, in the event the received information was not modified, evaluating the permission associated with the application, and in the event the permission is granted, executing the application.

#### BRIEF DESCRIPTION OF THE DRAWINGS

[0018] The accompanying drawings, which are incorporated in and constitute a part of the specification, illustrate presently preferred embodiments of the invention and, together with the general description given above and the detailed description of the preferred embodiments given below, serve to explain the principles of the invention. In the drawings:

[0019] Figure 1 is a flowchart depicting the high-level process of safe application distribution and execution in an exemplary embodiment of the present invention;

[0020] Figure 2 is a block diagram depicting a system architecture in which an exemplary embodiment of the present invention may be practiced;

[0021] Figure 3 is a block diagram depicting a wireless network architecture in which a safe application distribution processing system may be practiced in an exemplary embodiment of the present invention;

[0022] Figure 4 is a block diagram depicting a wireless device and some internal components in an exemplary embodiment of the present invention;

[0023] Figure 5 is a block diagram depicting information used to create a digital signature and transmitted to a wireless device in an exemplary embodiment of the present invention;

[0024] Figure 6 is a flowchart depicting the steps used to by a server or servers in distributing an application in an exemplary embodiment of the present invention; and

[0025] Figure 7 is a flowchart depicting the steps used by a wireless device when executing an application in an exemplary embodiment of the present invention.

### **DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT**

[0026] Reference will now be made in detail to the exemplary and preferred embodiments of the invention as illustrated in the accompanying drawings, in which like reference characters designate like or corresponding parts throughout the several drawings. The nature, objectives and advantages of the present invention will become more apparent to those skilled in the art after considering the following detailed description in connection with the accompanying drawings.

[0027] The present invention provides safe and secure application distribution and execution by providing systems and methods that test an application to ensure that it satisfies the predetermined criteria associated with the environment in which it will execute. Furthermore, by using rules and permission lists, application removal, and a modification detection technique, such as digital signatures, the present invention provides mechanisms to safely distribute and execute a tested, or untested, application by determining whether the application has been modified, determining if it has permission to execute in a given wireless device environment, and removing the application should it be desirable to do so.

[0028] It will be recognized to those skilled in the art that the forgoing describes an application file type being distributed and executed for simplicity of description. An "application" may also include files having executable content, such as: object code, scripts, java file, a bookmark file (or PQA files), WML scripts, byte code, and perl scripts. In addition, an "application" referred to herein, may also include files that are not

executable in nature, such as documents that may need to be opened or other data files that need to be accessed.

[0029] Figure 1 is a flowchart depicting the high level process of safe application distribution and execution in a manner consistent with an exemplary embodiment of the present invention. An embodiment of the present invention enables a developer identification to be associated with the application, performs testing on the application for the environment where the application is intended to execute, assigns permissions that can be used to dictate which devices or systems may execute the application, and provides for application removal should an application perform illegal or undesirable actions.

[0030] It is preferable that systems and methods employ all these techniques to increase the safe distribution and execution of an application. It will be recognized, however, that even employing one or more of these techniques will increase the safe distribution and execution of an application.

[0031] The high level process begins by associating the developer identification with an application (Step 100). This process may be performed by binding the developer identification with the application as it is distributed. Alternatively, the associated developer identification may be stored along with the corresponding application on a server in the system. It is also preferable that the developer identification information be stored and associated with the application information so that it cannot be easily modified.

[0032] The application is then tested for improper operation (Step 105). The application may be used in an environment where improper operation may not only affect the device on which the application is running, but also those other devices that are connected or networked with that device. It is preferable to test the application so that it does not make improper system calls or negatively affect the device or other connected devices during its operation. In one embodiment, this testing is performed by a certification process where the application is tested to determine if it meets predetermined criteria. It is preferable also, to have a certification process independent of the developer, to test the application. The independence of the certification process encourages more accurate and reliable testing.

[0033] Prior to executing the application, the application is checked to determine if it is “allowed” to execute on the device (Step 110). This check may be performed by the use of permissions and rules, described below, or by other permission mechanisms known to those skilled in the art. Furthermore, it is preferable that the application be checked prior to every attempt to execute the application. This persistent checking process increases the safety of executing the application. For example, it guards against an application having a Trojan horse that may have been inserted into that application on the executing device via another application.

[0034] An application that performs an improper or undesirable operation is then removed from the device (Step 115). This prevents the application from doing any further damage and also frees up the memory in the device for other uses. Alternatively, the application does not need to be removed from the application. Removing an application may refer to the disabling of the application and leaving the application on the device.

[0035] Figure 2 depicts a system architecture in which an exemplary embodiment of the present invention may be practiced. A developer 200 creates an application for use on the wireless device 230. As described above, it will be recognized by those skilled in the art that, while the foregoing description includes an application file type, other file types may be used. Furthermore, it will be recognized by those skilled in the art that the present invention may be used with other wireless or non-wireless devices and may employ wireless networks, non-wireless networks, or a combination thereof.

[0036] Typically, the developer 200 will have a set of development specifications for which to develop the application to execute on the wireless device 230. In one embodiment, the wireless device includes a software platform to assist the application’s interface with the wireless device, such as the BREW™ software developed by QUALCOMM, Incorporated, headquartered in San Diego, California. The developer may create the application satisfying the software platform, or BREW™ software, specification standards, and conventions.

[0037] The developer 200 is connected to a central server 205, in one embodiment, so that it may electronically transmit the application to the central server 205. In one embodiment, the central server is an Application Control Center Headquarters (ACCHQ)

server used in the distribution of applications to wireless devices. The developer 200 may digitally sign the application (discussed further below) to determine if the application was modified. It will be recognized that a physical connection to the central server is not necessary. For example the developer 200 may send, such as via first class mail, the application to the central server 205 stored on a CD-ROM.

[0038] In addition, the developer sends various source identification information to the central server 205. This source identification information may include any type of information that may be associated with the application that identifies the developer, such as a company name, tax identification of the company, or other identifying information.

[0039] The central server 205, either by itself or using a certification server 210, is used in the analysis and certification of applications. In one embodiment, an Application Control Center (ACC) may be used as a certification server. The certification server 210 may be used to analyze the application to determine whether the application satisfies predetermined certification criteria. This criteria may include whether the application satisfied the development specification for execution on a wireless device or platform. The certification criteria, however, can be any criteria that an application must satisfy prior to execution on a wireless device or platform. Such criteria may include verifying: (a) that the application functions as claimed by the developer, so that the application does not harm the wireless device's operation (e.g., it does not crash the phone); (b) that the application does not access data or memory that it should not (e.g., it doesn't access data or files owned by other applications, the operating system or platform software); and (c) that it does not negatively impact the wireless devices resources, such as detrimentally monopolizing the input and output of the wireless device.

[0040] The central server 205 may also assign a set of permissions in a list associated with the application. This permission list is determined by various factors, including an analysis of whether the application passed the certification process, what networks 220 the application is approved to execute on, and whether the wireless device supports the application. There may be many factors used to determine a permission list and is left to those skilled in the art when implementing the present invention.

[0041] The central server 205 receives the developer identification information and correlates it with the application created by the developer 200. Should any problems



exist with the application, the central server will be able to identify the source of the application. In one embodiment, the developer information is passed to the wireless device 230 so that the correlation may be performed by the wireless device or other systems connected to the wireless device.

[0042] In one embodiment, the central server is also connected to an application download server (ADS) 215. The application download server 215 is used to interface with a wireless device via a wireless network 220 to download an application. The central server may also send the permissions list and developer identification associated with the application to the ADS where it can be stored until transmission to a wireless device. It is preferred that the application, permission list and the developer identification be digitally signed by the central server to increase security from modification.

[0043] It will be recognized by those skilled in the art that an ADS may be used to connect to multiple networks 220 for the distribution of applications, files and other information to the various wireless devices 230. Furthermore, wireless and non-wireless networks may be employed to transmit the application's permission list and developer identification to the wireless device.

[0044] In response to a request for an application, the ADS 215 will send the application, permission list, developer identification, and digital signature(s) to the wireless device 230 via the network 220. In one embodiment, the wireless device 230 will contain a key to check the digital signature in order to determine if the application, permission list and/or developer information is modified.

[0045] It is preferred, if digital signatures are employed in the present invention, that the central server use a secure key to create the digital signature and install a key on a wireless device to evaluate the digital signature. By using a secure key, the wireless device will have a higher degree of reliability that the digital signature was created by the central server and not an imposter.

[0046] Should the application cause any errors on the wireless device, or for any other desired reason, the wireless device may initiate the removal of the application. Furthermore, an application may be removed from the wireless device based on a request from the ADS or central server. This request from the server may be initiated for any

desired reason. For example, the server may initiate the removal of applications from a wireless device because the application performed improperly on another device, a new version of the application is distributed, or even business reasons dictating that the application should be removed. This application removal process further protects the wireless device environment from repeated execution of corrupted and/or destructive applications.

[0047] Figure 3 depicts a wireless network architecture in which an application distribution system may be practiced in an exemplary embodiment of the present invention. A central server 302 is an entity that certifies, either by itself or in combination with a certification server, the application programs as compatible with a defined set of programming standards or conventions. As described earlier, these programming standards may be established so that the application will execute on a software platform, such as the BREW™ platform.

[0048] In one embodiment, the central server database 304 consists of a record of the identifications for each application program downloaded at any time onto each wireless device 330 in the network 300, an Electronic Service Number ("ESN") for the individual who downloaded the application program, and a Mobile Identification Number ("MIN") unique to the wireless device 330 carrying that application program. Alternatively, the central server database 304 contains records for each wireless device 330 in the network 300 of the wireless device model, wireless network carrier, the region where the wireless device 330 is used, and any other information useful to identify which wireless device 330 are carrying which application programs. In addition, the central server database may also store this developer identifying information associated with an application.

[0049] In one embodiment, the central server 302 may also include a remove command source 322. The remove command source 322 is the person(s) or entity(ies) that may make the decision to remove one or more targeted application programs. The remove command source 322 also is the entity that constructs a remove command 316 (discussed below) that is broadcast to identified wireless device 330 carrying the targeted application program(s). Alternatively, and without limitation, the remove command source 322 may be one or more persons or entities involved with the development and issuance of the targeted application program, persons or entities involved with the manufacturing of the

wireless device 330, and/or persons or entities involved with the function of any part of the network 300.

[0050] The central server 302 communicates with one or more computer servers 306, e.g., an ADS, over a network 308, such as the Internet, preferably secured. The servers 306 also communicate with a carrier network 310 via a network 308. The carrier network 310 communicates with the MSC 312 by both the Internet and Plain Ordinary Telephone System (POTS) (collectively identified in Figure 3 as 311). The Internet connection 311 between the carrier network 310 and the MSC 312 transfers data, and the POTS 311 transfers voice information. The MSC 312, in turn, is connected to multiple base stations ("BTS") 314. The MSC 312 is connected to the BTS by both the Internet 311 (for data transfer) and POTS 311 (for voice information). The BTS 314 sends messages wirelessly to the wireless devices 330 by short messaging service ("SMS"), or any other over-the-air method.

[0051] One example of a message sent by the BTS 314 in the present invention is a remove command 316. As further discussed herein, the wireless device 330, in response to receiving a remove command 316, responds by uninstalling a targeted application program stored on the wireless device 330. In one embodiment, the remove program may additionally or alternatively be programmed to disable the targeted application program or reprogram it to perform differently. The wireless device may also delete the application and any related information, such as a permission list.

[0052] The remove command 316 is constructed by the remove command source 322 (which may or may not be the same person(s) or entity(ies) that made the decision to initiate a remove of the targeted application program). The remove command 316 is sent by the remove command source 322 over the network 300 for broadcasting to the wireless devices 330.

[0053] By using the remove command as described in the above embodiment, the safety of application distribution and execution is increased by providing a mechanism to uninstall corrupted or undesirable applications. It will be recognized by those skilled in the art that, while the preceding described a remove command initiated by the central server, the wireless device may also initiate the removal or uninstallation of the application and its related information.

[0054] Similarly, the above network may be used to send the application, permission list and associated digital signatures from the central server to various servers 306 (e.g., ADS') through the MSC and BTS to the wireless devices 330.

[0055] Figure 4 depicts a wireless device and some internal components in an exemplary embodiment of the present invention. While this embodiment is directed to a wireless device 400, it is used as an example without any intended limitation. The invention could alternatively be performed on any form of remote module capable of communicating through a network, including, without limitation, wireless and non-wireless devices, such as personal digital assistants ("PDAs"), wireless modems, PCMCIA cards, access terminals, personal computers, devices without a display or keypad, or any combination or sub-combination thereof. These examples of remote modules also may have a user interface, such as a keypad, visual display or sound display.

[0056] The wireless device 400 shown in Figure 4 has an application-specific integrated circuit ("ASIC") 415 installed at the time the wireless device 400 is manufactured. The ASIC is a hardware component that is driven by software included in the ASIC. An application programming interface ("API") 410 is also installed in the wireless device 400 at the time of manufacture. In one embodiment, the API represents a BREW API or software platform. The API 410 is a software program configured to interact with the ASIC. The API 410 serves as an interface between the ASIC 415 hardware and application programs (discussed below) installed on the wireless device 400. Alternatively, the wireless device 400 may contain any other form of circuitry that would allow programs to be operated in a manner that is compatible with the hardware configuration of the wireless device 400. The wireless device 400 also has storage 405. The storage 405 consists of RAM and ROM, but, alternatively, could be any form of memory, such as EPROM, EEPROM or flash card inserts.

[0057] The storage area 405 of the wireless device may be used to store received applications and permission lists 425. In addition, the storage area 405 may be used to store one or more "keys" 405. These keys can be applied to a digital signature using a signature algorithm to determine whether the signed information was modified.

[0058] Rules 435 may also be installed on the wireless device 400. These rules may be used in conjunction with the permission list to determine if an application is allowed to

execute. For example, a rule may state that an application is allowed to execute if a certification flag is set in the permission list (i.e., indicating the application passed certification). The permission list will have the certification flag set or not, depending on whether it passed certification. By applying the rule to the information contained in the permission list, permission to execute the application is either granted or denied.

[0059] The manufacturer (not shown) of the wireless device 400 may download application programs onto the storage 405 of the wireless device 400 at the time the wireless device 400 is manufactured. These application programs may be any program potentially useful or entertaining to the user of the wireless device, such as games, book, or any other type of data or software programs. The application programs also may be downloaded onto the wireless device 400 over the air after the wireless device is manufactured.

[0060] The remove program, when executed by the wireless device 400, uninstalls one or more targeted application programs from one of the applications stored on the wireless device 400. The targeted application program is an application program that needs to be uninstalled from the wireless device 400 for various reasons discussed below.

[0061] The wireless device 400 has a local database 420 installed by the manufacturer. The API of the wireless device is programmed to automatically update the local database 420 with a record of identifying information about each of the application programs stored on the wireless device 400. The local database 420 contains a record of the signature identifications unique to each application program stored on the wireless device 402. Additionally, the local database 420 may contain a record of the location of the application programs within the storage 405 on the wireless device 400 and any other information useful for keeping track of which application programs are downloaded on the wireless device 400, and where they are located.

[0062] Figure 5 is a block diagram depicting information used to create a digital signature and transmitted to a wireless device in an exemplary embodiment of the present invention. As is known by those skilled in the art, digital signatures may be used to track whether a digital file has been modified. As described, digital signatures can be applied to any digital file, including documents, applications, databases, etc. In general, a digital signature is created by applying a key to a file using a signature algorithm. This digital

signature is created using the information contained in the file. Typically, the digital signature is sent along with the file to a recipient. The recipient of the file and digital signature can then apply a key to the received file and digital signature to determine if the file has been modified during the transmission to the recipient.

[0063] Keys used to create and evaluate a digital signature can be used to determine the identity of the signer. For example, a key may be generated to create a digital signature by an entity and kept securely. This entity can distribute a corresponding key that can be used to evaluate the digital signature. If the key is kept securely and not compromised, the recipient evaluating the digital signature can determine not only whether the information was modified, but also the identity of the signer.

[0064] Alternatively, third-party entities can create keys for specific entities in a secure fashion. Therefore, a recipient having a key associated with a specific identity will be able to determine if that entity was the signer.

[0065] In one embodiment of the present invention, a digital signature 515 is generated by using the signer's key 525, e.g., a key of the central server (see Figure 2), the application 500, the permission list 505 and the developer identity information 510 as input to a digital signature algorithm 530. The result is a digital signature 515 that is dependent on the information contained in the inputs.

[0066] After creating the digital signature 515, the application 500, permission list 505, developer identity information 510 and digital signature 515 are transmitted to the wireless device 520. The wireless device can then use the digital signature to determine if any of the application or related information (i.e., the permission list and developer identity information) was modified. In addition, using one of the techniques described above, such as a secure key, the wireless device may also have confidence in the identity of the signer who transmitted this information to the wireless device.

[0067] Figure 6 is a flowchart depicting the steps used to by a server or servers in distributing an application in a manner consistent with an exemplary embodiment of the present invention. In this exemplary embodiment, the process begins by receiving an application and a digital signature (Step 600). The digital signature is information related to the application such that it can be determined if the application was modified prior to its reception. Furthermore, it is preferable that the key used to sign the digital signature

be assigned by a third party in order validate that the entity or developer signing the application is the developer that received the assigned key.

[0068] After receiving the application and digital signature, the digital signature is evaluated to determine if the developer who sent the application is the same as the one who signed the application (Step 605). If a third party assigned the key to the developer to create the digital signature, then the third party may also assign the key to evaluate the digital signature to the receiving party, such as to the central server described with respect to Figure 2.

[0069] The identification of the developer, or whichever entity signed and/or created the application, is then stored and associated with the application (Step 610). The storage may be in a table, database or in some other manner such that it can be later retrieved in the event the identity of the developer needs to be determined. In one embodiment, the storage of the developer's identification is stored in a wireless device and not in a server.

[0070] The received application is then certified to determine if it meets specified criteria (Step 615). In one embodiment, an application may be written to execute on a specific platform, such as the BREW™ platform developed by QUALCOMM, Incorporated, headquartered in San Diego, California used in wireless devices. A specific platform, or device, may have specific requirements that an application must meet prior to having it executed on the device. For example, a platform or device may require that an application not access specific memory locations in the device so that the integrity of the device or other applications located in memory are not compromised. These criteria can be specified and the application can be tested to determine if these criteria are met. Preferably, these criteria are predetermined and provided to the developer to incorporate into the application's development.

[0071] After certification, the permissions associated with the application for a given environment are assigned (Step 620). Permission may be assigned based on many factors, depending on the environment in which the present invention is implemented. In one embodiment, the applications are intended for a wireless device. In this embodiment, assigning permissions may depend on the carrier network, a wireless device's requirements, results of certification testing, and developer, carrier or other testing environments, for example. Therefore, an example of a permission list is an indication

that the application passed certification testing and that it may execute on a specific carrier's network.

[0072] The server then digitally signs the application, permission list, and developer identification (Step 625). In one embodiment, this signature is performed using a secure key so that the identity of the server can be determined by those receiving this digitally signed information. It is not required that the developer's signature that was received by the server also be signed or that the developer's signature be sent to the wireless device.

[0073] The application, permission list, developer identification and the signature created in step 625 is then transmitted to a wireless device (Step 630).

[0074] Figure 7 is a flowchart depicting the steps used by a wireless device when executing an application in a manner consistent with an exemplary embodiment of the present invention. In this embodiment, the wireless device stores rules to evaluate permissions associated with an application (Step 700). It will be recognized by those skilled in the art that, while the present invention describes a rules/permissions paradigm, there are many paradigms that may be used to grant permission to an application for a specific device or platform and these are considered within the scope of the present invention.

[0075] The wireless device then receives the application, permission list, developer identification and digital signature (Step 705). In one embodiment, the wireless device may evaluate the received digital signature to determine the identity of the signer. The digital signature may also be used to determine if the application, permission list or developer identification was modified after having been signed.

[0076] The wireless device then receives a request to execute the application (Step 710). This request may come from the user of the wireless device wanting to execute a program. Alternatively, the request may be made by the wireless device itself or from some request transmitted to the wireless device, either through a network or direct connection to the wireless device.

[0077] After receiving the request, the wireless device evaluates the digital signature and the permission list associated with the application prior to its execution (Step 720). As described, the wireless device, in one embodiment, may use rules to evaluate the permissions list. If by evaluating the digital signature it is determined that the



application, permission list or developer identification was not modified, then the wireless device evaluates the permission list using the stored rules. If there was no modification and the evaluation of the rules against the permission list indicates that the application is granted permission to execute in the wireless device, the processing proceeds to execute the application on the device (Step 730).

[0078] If the evaluation in Step 720 indicates that either the application, permission list or developer identification was modified after being signed, or that the application is denied permission to execute on the wireless device, then the application is not executed (Step 725). Processing proceeds to remove the application from the wireless device (Step 750). It is also preferred that the permission list and developer identification also be removed from the wireless device.

[0079] Following step 730, the application's execution is monitored to determine if it performs an illegal or improper operation (Step 735). The wireless device or the platform the wireless device is using may define certain operations to be illegal or improper. These operations may include those that access restricted areas of memory or memory locations used by other programs or files. In addition, these operations may involve harmful uses of the wireless device's resources such that they may not only affect the wireless device, but other devices on the network the wireless device is attached.

[0080] If such an illegal or improper operation is attempted, then the application's execution is stopped (Step 745) and removed from the wireless device (Step 750) along with, preferably, the developer identification and permission list. As stated above, alternatively, the remove process may involved the disabling of the application, thereby preventing its execution, and keeping the application on the wireless device.

[0081] If no illegal, improper, or undesirable operation is performed in step 735, then the application is allowed to continue execution (Step 740).

## CONCLUSION

[0082] Using mechanisms to certify, detect modifications, determine source identity, assign permissions, and incorporate the ability to remove the application, systems and methods consistent with the present invention increase safe and secure application distribution and execution. Systems and methods may implement as few or all of these

mechanisms. The more mechanisms implemented, the higher the degree of safety that is achieved.

[0083] In one embodiment, a developer sends an application to a server. The developer may sign the application to protect against unauthorized modification. A server checks the identity of the developer, and performs certification testing on the application. The server also assigns permissions to the application, creating a permission list. The application, permission list, developer identification are digitally signed by the server and sent to a wireless device along with the digital signature. A wireless device checks the digital signature for modification and the permission list against stored rules prior to executing the application. In one embodiment, these checks are performed prior to each attempt to execute the application on the wireless device. If the checks indicate the application has been modified or denied permission to execute, the application does not execute and is removed from the wireless device. Furthermore, if during execution, the application attempts an illegal or improper operation, the application is terminated and then removed from the wireless device.

[0084] The foregoing description of an implementation of the invention has been presented for purposes of illustration and description. It is not exhaustive and does not limit the invention to the precise form disclosed. Modifications and variations are possible in light of the above teachings or may be acquired from practicing of the invention. For example, the described implementation includes software but one embodiment of the present invention may be implemented as a combination of hardware and software or in hardware alone. The invention may be implemented with both object-oriented and non-object-oriented programming systems. Additionally, although aspects of the present invention are described as being stored in memory, those skilled in the art will appreciate that these aspects can also be stored on other types of computer-readable media, such as secondary storage devices, like hard disks, floppy disks, or CD-ROM; a carrier wave from the Internet or other propagation medium; or other forms of RAM or ROM. The scope of the invention is defined by the claims and their equivalents.